SVT H/L programming

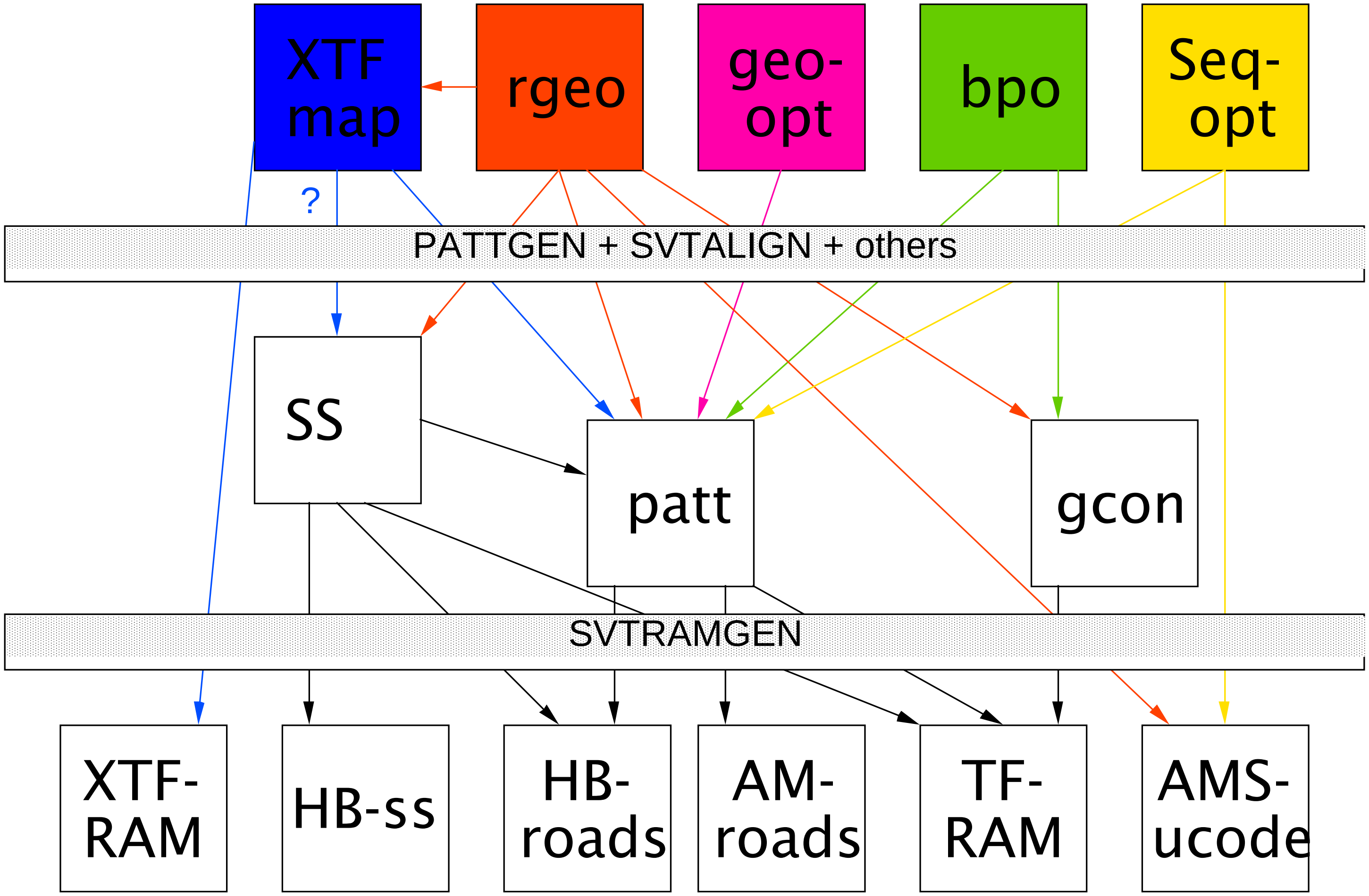
Giovanni Punzi

SVT software workshop 2/5/01

What I am talking about

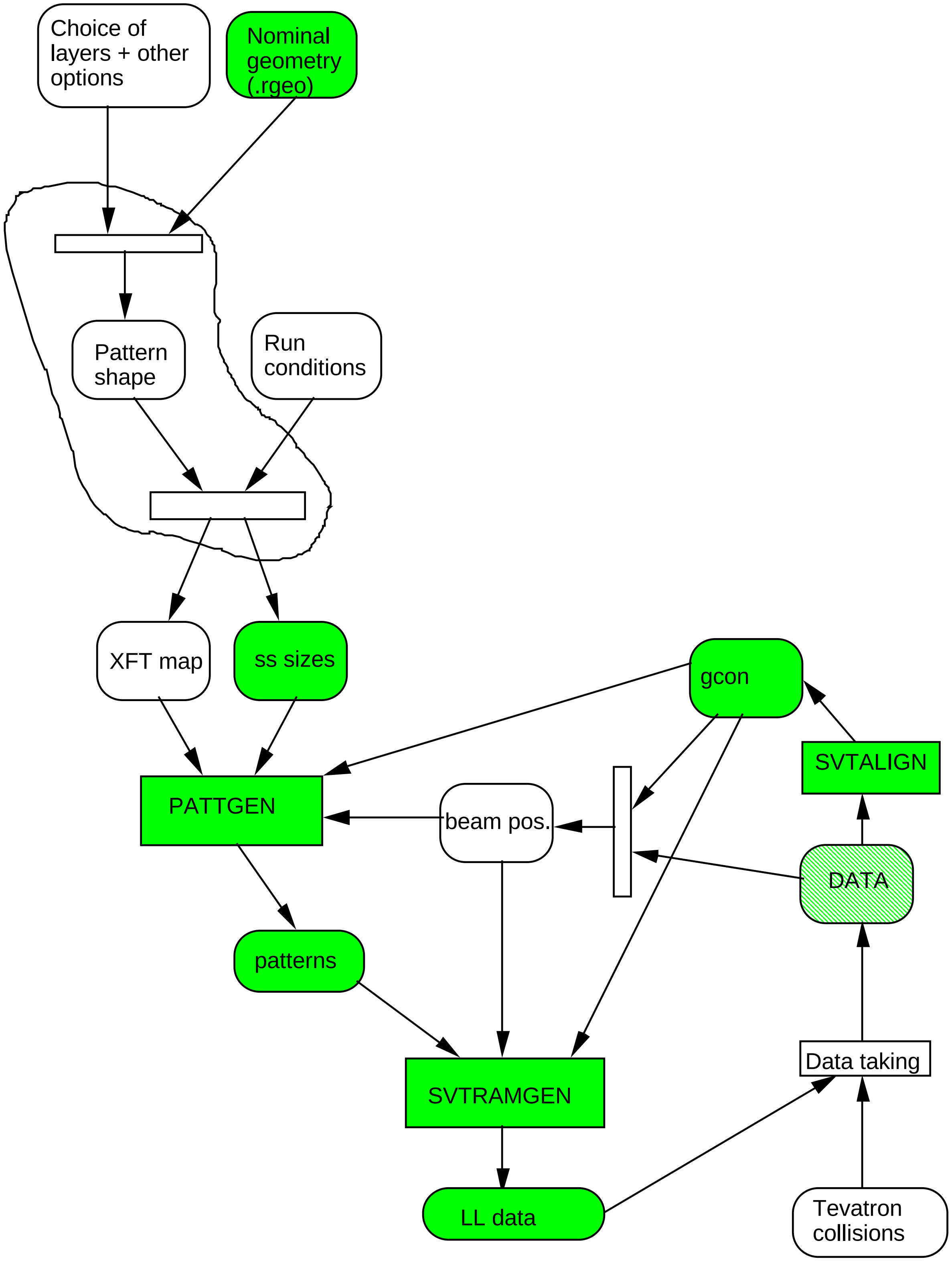
- Organization of data for programming SVT at a high level
- Has to do mostly with *patterns* and *fit constants*
- Interacts with a number of other things
- Requirements:
  - Depends on run conditions: generate new files during data taking
  - Handle storing/retrieving from DB
  - "Compile" into binary data

File dependencies



Giovanni Punzi - Pisa

Giovanni Punzi - Pisa



Files

- Patterns (.patt):
  - generate on demand
  - O(1Mb/sector) make a more compact form ? Go into DB.
  - Incorporate ss info ?
  - ~1 CPU hour to generate
- Fit constants (.gcon)
  - Small, generate from data samples
  - ~minute of CPU to generate
  - Go into DB

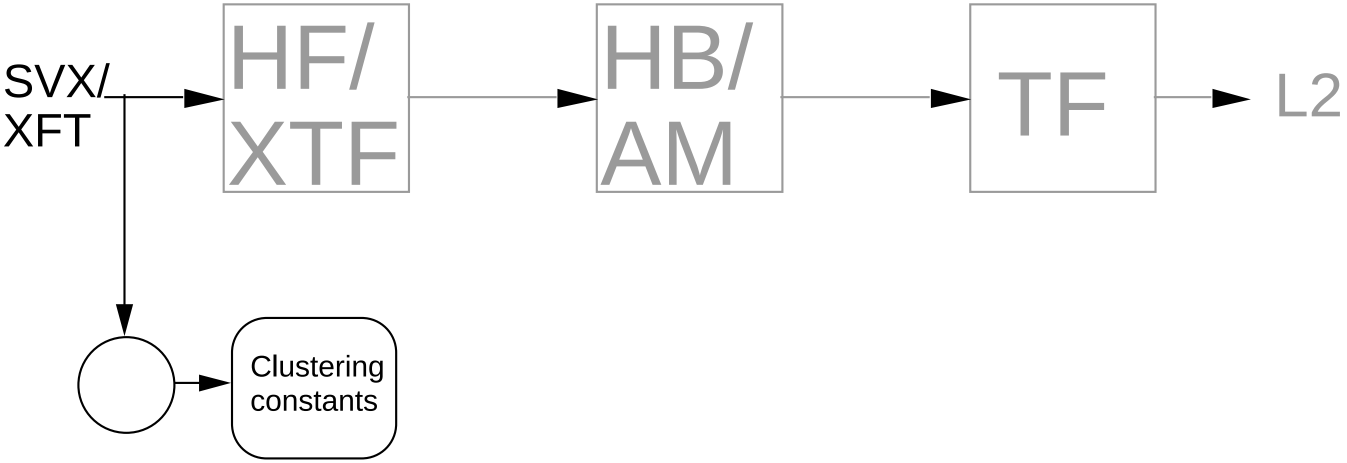
Files

- HF ? (apart from few constants, will come from SVX DB)
- XTF map ?
    - New, still working on it
    - Large, but uses few high-level parameters …currently we use it to tranform phi6 and pt into phi at some point close to SVX
    - More sophisticated functions can be implemented to give larger efficiency
    - May want to store full map for flexibility
- Pattern shapes: few of them, pre-calculate all
- SS sizes (.ss): few of them, pre-calculate all

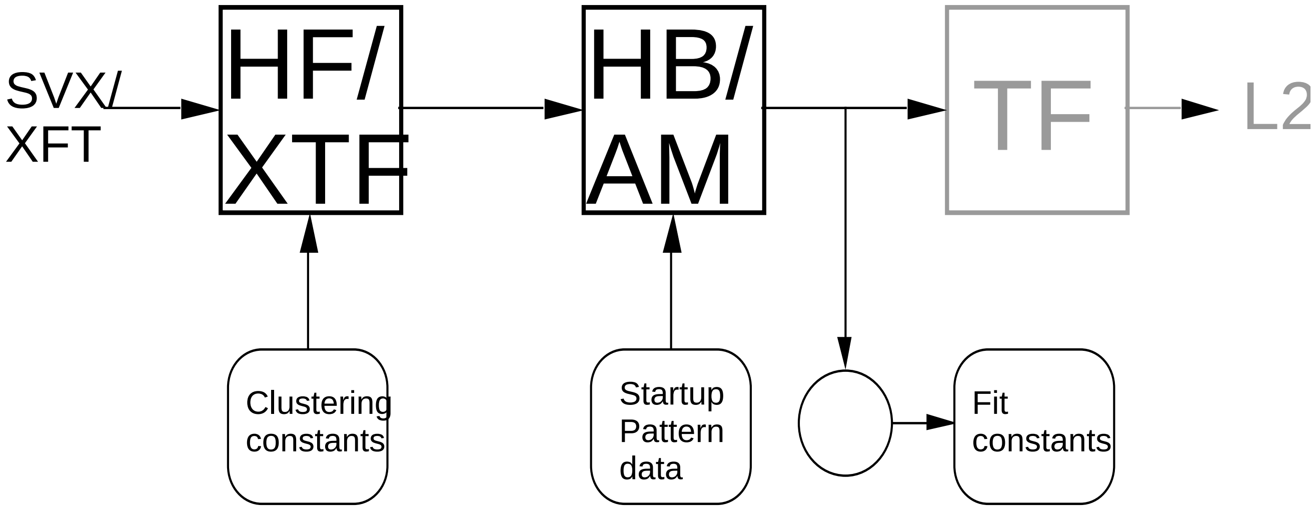
Interfacing

- Those files are needed in several places
- I/O through standard routines -> C structures
  - Are they OK for every other use ?
- The code is in cvs, several programs already use it
- Formats, both internal and external, need web documents
- Make everything sector-oriented ? I suggest it.
- Need to do something similar for DB I/O.
  - Big issue: how is this going to work ?

Startup step 1:pedestal/hot



Startup step 2: alignment



SVX/
XFT

HF/
XTF

HB/
AM

TF

L2

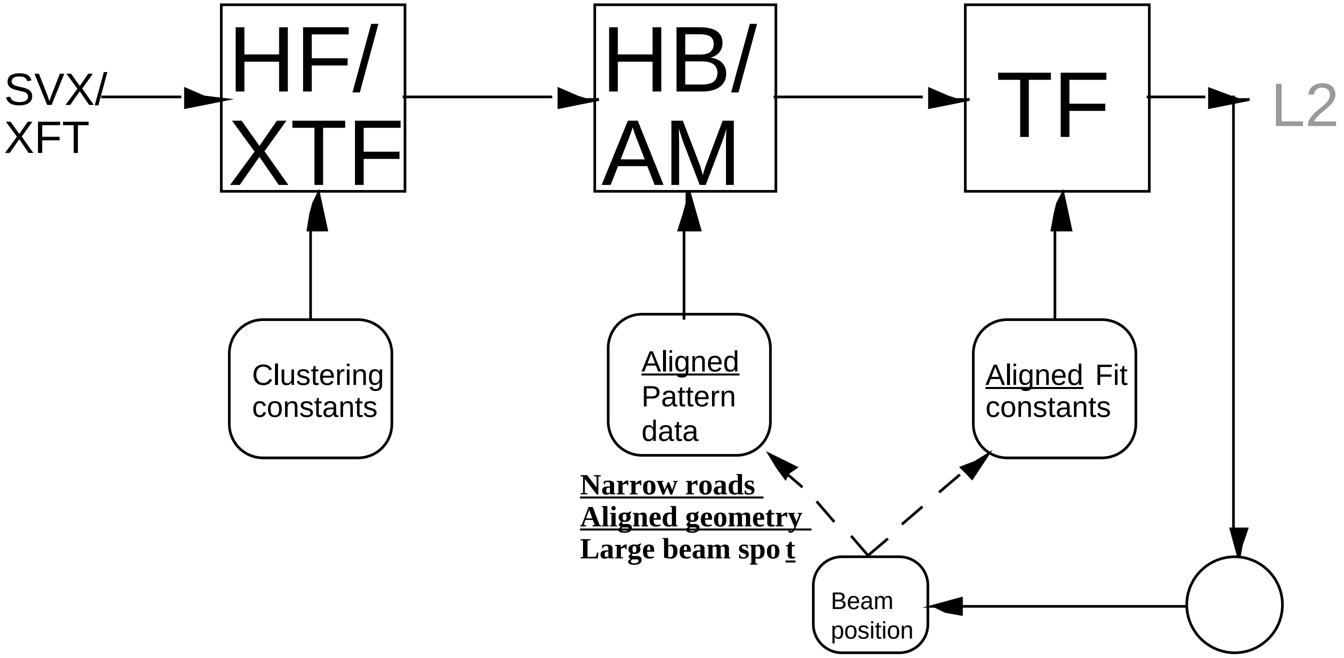
Clustering
constants

Startup
Pattern
data

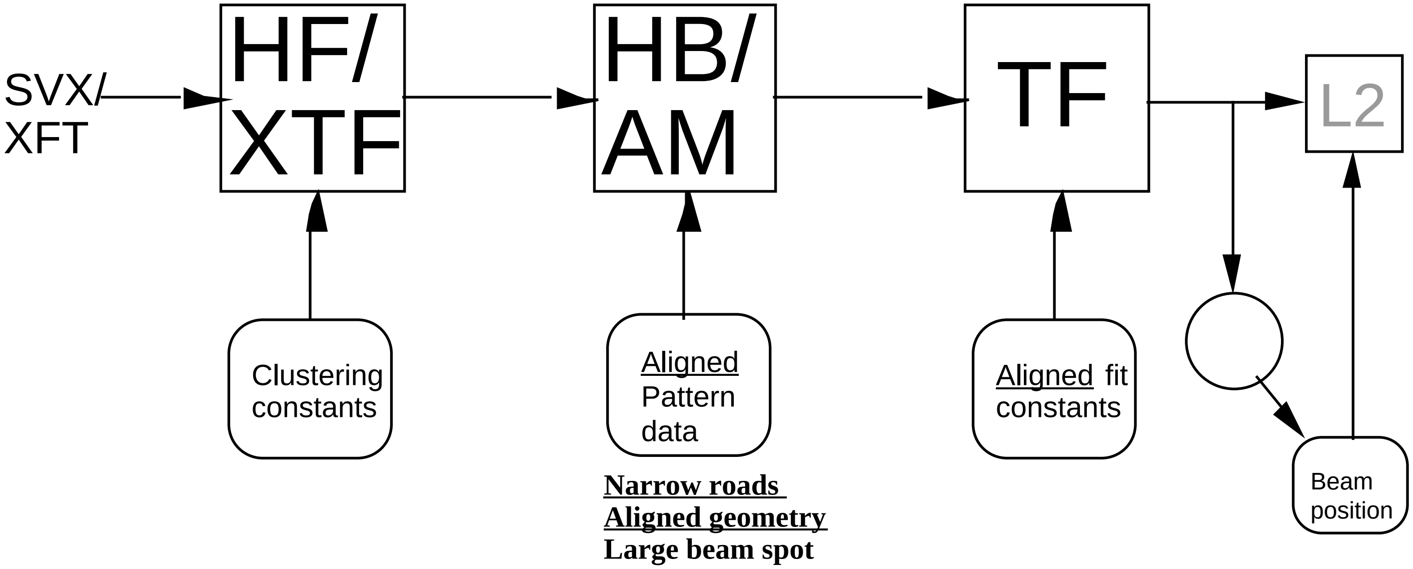
Fit
constants

Wide roads
Nominal geometry
Large beam spot

Startup step 3: Initial beam



Correcting for small beam shifts



Beam correction in L2, to be recorded in SVTD ?

Giovanni Punzi - Pisa

How it works in practice

- Proposal: do all work under online cvs, then export useful releases to offline cvs.
- To use, log into svt account
- Checkout/build needed software from cvs (svt_config)
- Set various options (via web forms)
- Get real data samples to align
- Run to make files
- Store them into DB

Directory organization

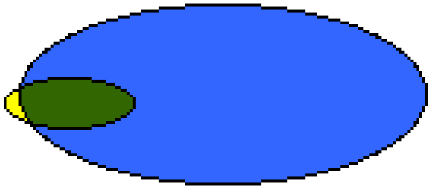
- svt_config
  - source
    - <common code: I/O routines, header and the rest>
    - PATTGEN (pattern generator)
    - CORRGEN (nominal gcon generator)
    - SVTALIGN (makes aligned gcon)
    - SVTRAMGEN (library of routines for generating low-level files)
    - QWIKSVT (fast approximate simulator)
  - data
    - <all base files (pattern shape, ss etc)>
    - <all nominal files for startup>
    - Individual directories for specific configs ?

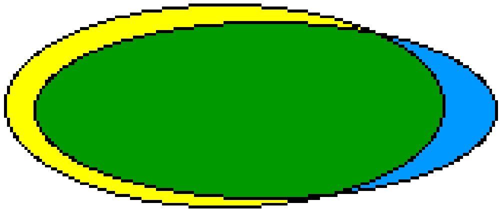
Most recent changes

- Changed ss files
- Changed conventions to be like offline
- Tested SVTALIGN on commissioning data

Alignment on commissioning

- Comparison, not normalized:
  - wedge 2:
    - (align over nom) = 0.163600
    - (nom over align) = 0.918600
  - wedge 3
    - (align over nom) = 0.223600
    - (nom over align) = 0.923200
- Comparison, normalized:
  - wedge 11
    - (align over nom) = 0.797800
    - (nom over align) = 0.800600
  - wedge 12
    - (align over nom) = 0.817600
    - (nom over align) = 0.823600





Issues

- Handling everything in an orderly way
- XTF map format
- Making everything sector-oriented
- More compact format for patterns ?
- How beam position fits in the scheme ?
- Storing/retrieving to/from DB

Plan

- Continue interaction with Bill
- Make all base files
- Generate all nominal files
- Define XTF map
- Make it easy to manage configs (web forms ?)
- Make appropriate interface for SVTALIGN and release it